THIS MANUAL IS CURRENT
FOR MDBS 3.09

MDBS III

INTERACTIVE DATA MANIPULATION LANGUAGE

REFERENCE MANUAL

VERSION 3.09

MARCH 1987

Micro Data Base Systems, Inc.

P. O. Box 248

Lafayette, Indiana 47902

USA

Telex: 9102401559(MDBS WLAF UQ)

(800) 344-5832

Copyright Notice

This entire manual is provided for the use of the customer and the customer's employees. The entire contents have been copyrighted by Micro Data Base Systems, Inc., and reproduction by any means is prohibited except as permitted in a written agreement with Micro Data Base Systems, Inc.

© Copyright 1987 Micro Data Base Systems, Inc.

3253924

NEW VERSIONS, LICENSING, AND DISCLAIMER

Any programming endeavor of the magnitude of the MDBS III software continually evolves over time. Realizing this, Micro Data Base Systems, Inc. (**mdbs**), provides updates to this product as part of a service and maintenance program.

New versions of MDBS III software are considered separate products; however, bona fide owners of previous versions are generally entitled to a preferential rate structure.

Each copy of our software is personalized to identify the licensee. There are several levels of this personalization, some of which involve encryption methods guaranteed to be combinatorially difficult to decipher. Our products have been produced with a very substantial investment of capital and labor, to say nothing of the years of prior involvement in the data base management area by our principals. Accordingly, we are seriously concerned about any unauthorized copying of our products and will take any and all available legal action against illegal copying or distribution of our products.

All rights reserved. No part of this material shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from **mdbs**.

Although care has been taken in the preparation of this material, **mdbs** assumes neither responsibility for errors or omissions, nor for damages resulting from the use of the information contained herein. **mdbs** does not warrant its accuracy nor guarantee the operation of the system in every instance described herein.

The reader/user assumes full liability and all risk for any damages resulting from the use of the information contained herein, and for determining whether the information contained herein is suitable for user's intended purpose.

mdbs reserves the right to incorporate design improvements and new functions in its software products and software systems. Recent improvements may not always be reflected in documentation.

© Copyright 1987 Micro Data Base Systems, Inc.

TABLE OF CONTENTS

Page

CHAPTER I. OVERVIEW I-1
Using IDML I-1
IDML Commands I-2
Notational Conventions I-4
On-Line Help I-4

CHAPTER II. DISPLAY COMMAND II-1

CHAPTER III. INVOKING DML COMMANDS INTERACTIVELY III-1

CHAPTER IV. DEFINE COMMAND IV-1
Advanced Usage of Macros with Parameters IV-5

CHAPTER V. READ COMMAND V-1

CHAPTER VI. ECHO COMMAND VI-1

CHAPTER VII. COMMENT COMMAND VII-1

CHAPTER VIII. SET COMMAND VIII-1

CHAPTER IX. QUIT COMMAND IX-1

CHAPTER X. STARTUP FILE X-1

CHAPTER XI. MESSAGES XI-1

Figure I-1. Sample MDBS III Schema I-3

APPENDIX A. DML Command Groups A-1

APPENDIX B. DML Command String Forms B-1

© COPYRIGHT 1987 Micro Data Base Systems, Inc.

I. OVERVIEW

The Interactive Data Manipulation Language (IDML) is an approach to accessing MDBS III data bases. Alternative access approaches include the DML, QRS, and RDL. IDML is similar to QRS in that it is interactive, independent of any host language, and supports several QRS commands (such as DISPLAY, READ, and user-defined macro commands). IDML is similar to the DML in that it supports most of the DML commands presented in the MDBS DMS Manual. With IDML, a user can execute each of these DML commands one-at-a-time. If a DML command requires data inputs, then the IDML processor prompts the user for appropriate data values. If a DML command extracts data from a data base, then the extracted values are displayed at the console.

IDML is particularly valuable as an aid in learning and testing DML logic. It is also valuable for quickly making low volume, ad hoc changes to data values and their interrelationships, without writing a program. All data integrity and security conditions as specified in the DDL are automatically enforced by IDML. Synonyms as defined with the DDL are generally recognized by IDML.

USING IDML

Simple IDML installation and execution instructions are included in the pertinent system specific manual. Once IDML has been installed, it can be executed from the operating system. IDML can be invoked with a -b argument on the command line to explicitly allocate the page buffer region. If this argument is not specified, approximately half of the available memory is automatically allocated. The remainder of available memory is reserved as a non-data base working space (e.g., file control blocks, stack, sort work areas). If the argument is used, it has the form -bnnnnn where nnnnn is the (decimal) number of bytes to be allocated for the page buffer region. This number should be at least as large as the "minimum DMS buffer region size" reported by the DDL Analyzer; otherwise DMS command status error 31 results. If too large a page buffer region is requested, an error message indicating "excessive memory request" is displayed. The remaining memory reserved for non-data base working space is normally sufficient. However, in rare cases this working space may be insufficient, resulting in various kinds of error conditions and situations where the data base may be left open. In such a case, -b can be used to allocate a smaller page buffer region to allow a larger non-data base working space.

In the operating system command line, the user can also optionally specify -l followed by the name of the transaction log file to be used during the IDML execution (see the pertinent system specific manual). The disks containing the main data base area and the log file must be on-line.

When the IDML processor is executed, an IDML banner message appears on the console screen. This banner can be suppressed by including a -m argument in the command line used to invoke the IDML processor. In any event, the user is prompted for the file name of the main data base area, user name, and user password. These prompts can be suppressed by including -d, -u and -p arguments on the command line used to invoke IDML. If used, the -d should be followed immediately by the file name. Similarly, -u and -p should be followed by the user name and password respectively.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Overview

IDML checks to see whether the user name and password are valid for the indicated data base. If they are valid, then the I: prompt is issued and the data base is automatically opened. This means that IDML is ready to accept an IDML command. An IDML command is terminated by pressing the carriage return key. After IDML finishes its response to command, the I: prompt is again issued to show that IDML is ready to accept the next command.

Execution of IDML is terminated by entering either QUIT or BYE. This automatically closes the data base. This returns control to the operating system.

Chapter X explains how to create and make use of a STARTUP file, which IDML can automatically utilize to identify the data base and user.

IDML COMMANDS

The Interactive Data Manipulation Language supports a variety of standard commands. In addition, a user is allowed to define still other commands (called macro commands). The standard commands supported in IDML have the following effects:

- HELP: provide on-line descriptions of IDML command usage
- DISPLAY: display selected data dictionary contents
- DML commands supported:

AMM	FLO	GFC	ODRK	PFC	SOC
AMO	FMI	GFM	OFM	PFM	SOE
AOM	FMSK	GFO	OFO	PFO	SOM
AOO	FNM	GMC	OLM	PUTC	SON
AUI	FNMI	GOC	OLO	PUTM	SOO
CCU	FNMSK	GTC	OMI	PUTO	SOU
CRA	FNO	GTM	OMSK	RMS	SUC
CRS	FNOI	GTO	ONM	ROS	SUM
DBCLSA	FNOSK	IMS	ONMI	RSM	SUN
DBCNV	FNS	IOS	ONMSK	RSO	SUO
DBENV	FOI	LGFLSH	ONO	SCD	SUU
DBOPNA	FOSK	LGMSG	ONOI	SCM	TCN
DBSAVE	FPM	MAU	ONOSK	SCN	TCT
DBSTAT	FPMI	MCC	OOI	SCO	TMN
DRC	FPMSK	MCF	OOSK	SCU	TMT
DRM	FPO	MCP	OPM	SDC	TON
DRO	FPOI	MRID	OPMI	SMC	TOT
FDRK	FPOSK	MRTF	OPMSK	SME	TUN
FFM	FRK	MRTP	OPO	SMM	XMM
FFO	GETC	MSF	OPOI	SMN	XMO
FFS	GETM	MSP	OPOSK	SMO	XOM
FLM	GETO	NCI	ORK	SMU	XOO

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Overview

These commands have the effects described in the MDBS DMS Manual. Multiuser DML commands will have no effect if a multiuser version of the IDML processor is not being used. Logging commands have no effect if the RTL form of IDML is not being used.

- DEFINE: define macro commands
- READ: read and execute the IDML commands on a disk file
- ECHO: echo a message to the console screen
- COMMENT: set the symbol used to denote a comment
- SET: adjust the environment settings
- FILLCHR: set the symbol to be used as a fill character during data entry
- QUIT: stop processing IDML commands

These commands are described in the chapters that follow. The commands are presented in the order shown above. Figure I-1 shows the sample logical structure presented in Chapter II of the MDBS DDL Manual (in Figures II-3 and III-1,2). One additional set (ISKL) has been added to this schema. All examples in this manual are based on the logical structure of Figure I-1.

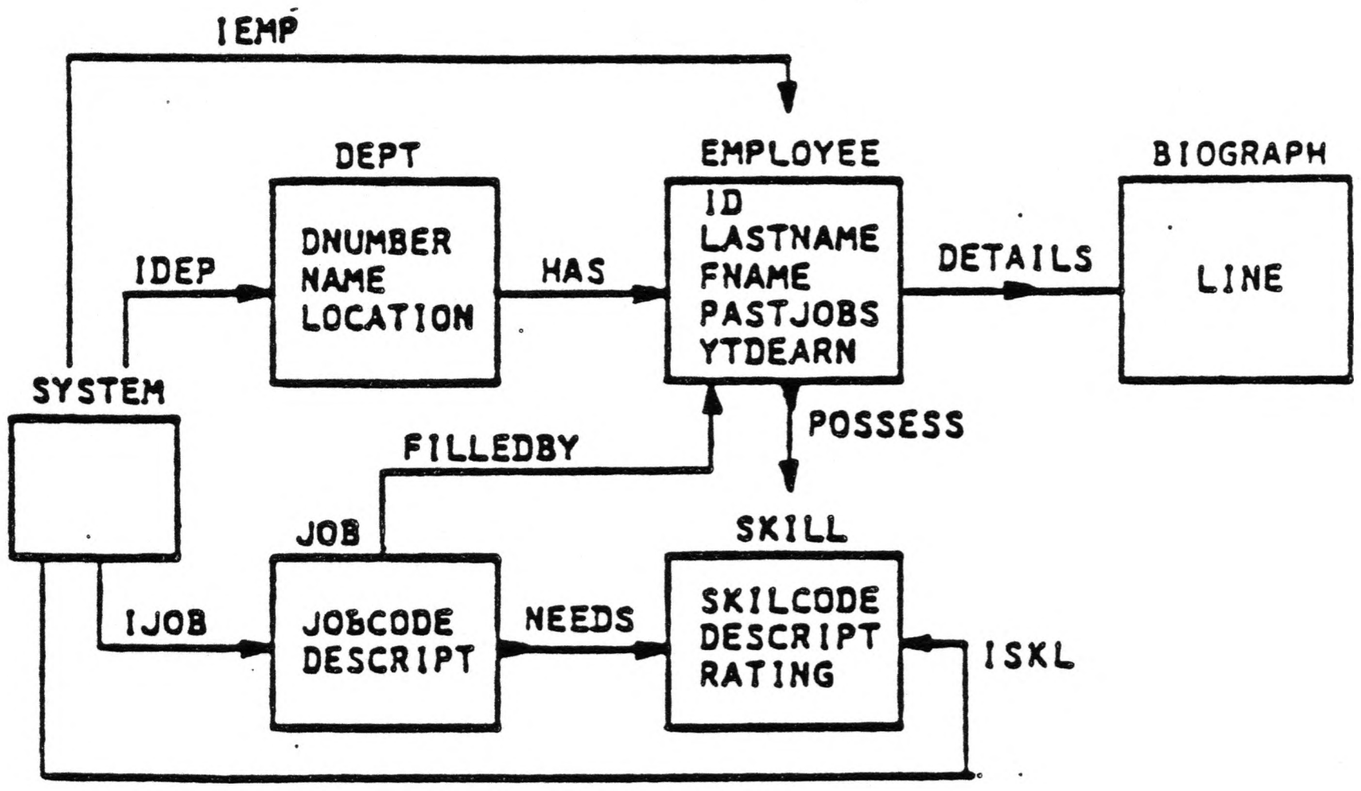

Figure I-1. Sample MDBS III Schema

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Overview

NOTATIONAL CONVENTIONS

When describing the syntax of an IDML command, each clause is enclosed within the symbols < and >. These two symbols are not used to separate clauses within an IDML command itself. In actually stating an IDML command, two clauses in a command can be separated by either blanks or a comma.

For purposes of syntactic documentation, keywords and other literals are indicated with bold faced type. User-determined elements of a command are not in bold faced type. For example, the display command has the syntax:

<display><*parameter*><*list*>.

A display command must literally begin with the word display; the user determines the parameter and list that are to be used in a particular display command.

ON-LINE HELP

IDML provides an on-line help facility. To get a display of helpful information about IDML commands, type the word **help** in response to the I: prompt. Command syntax is displayed using notational conventions similar to those described above.

Command syntax for an individual DML command can be seen by typing the command name followed by a question mark (?) symbol or help followed by the command name.

© Copyright 1987 Micro Data Base Systems, Inc.

II. DISPLAY COMMAND

SYNTAX

<display>
or
<disp>
or
<display><*parameter*><*list*>
or
<disp><*parameter*><*list*>

SYNOPSIS

The DISPLAY command lets you obtain information about the IDML environment and the schema of the data base. The schema information source is the data dictionary. You can retrieve information such as language character set, areas, record types, sets, or data items. If the data base has been initialized with the **o** option (see **MDBS DDL Manual**, Chapter VI), the schema information is proprietary and the DISPLAY command will not obtain data dictionary information.

DESCRIPTION

The simplest method to invoke the display command is to enter one of the following lines from the IDML prompt:

DISP

DISPLAY

This action brings you to the DISPLAY command menu, providing the various kinds of information accessible with DISPLAY.

The following display options are available:

(a) display areas
(i) display items
(l) display language type
(o) display option settings
(r) display records
(s) display sets
(q) quit display facility

Enter desired display option.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Display Command

The **a**, **i**, **l**, **r**, and **s** options retrieve virtually all schema information for the data base. Some security information, however, is inaccessible. The DISPLAY command does not retrieve the following items for security reasons:

- user names
- passwords
- user access codes
- access codes for data base elements

Read and write access codes for areas, data items, record types, or sets are not displayed. However, DISPLAY reports the read and write accesses of the current user.

The **o** option displays the environment settings available in IDML. The table lists each environment option, its status, and its meaning.

OPTION	STATUS	MEANING IF ON
5	OFF	echo alternate input
11	OFF	suppress macro recognition
12	OFF	echo macro expansion output
15	OFF	pause after each report page

Command Arguments

As an alternative to the menu system above, you can issue the DISPLAY command with one the following parameters to specify the information of interest:

<parameter>	information displayed about
a	display areas
i	display data items
l	display language used for sorting
o	display option settings
r	display record types
s	display sets
q	quit display facility

The **a, i, r** and **s** parameters accept a list to restrict the displayed information to a particular area, data item, record type, or set.

<parameter>	<list>	function
a	area name	display information about the indicated area
i	record type name,data item name	display information about the item in the indicated record type
r	record type name	display information about the indicated record type
s	set name	display information about the indicated set

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Display Command

The Escape key interrupts the DISPLAY command response to the screen. IDML returns its I: prompt and waits for the next command.

EXAMPLES

DISPLAY R	displays summary information for all record types
DISPLAY R SKILL	displays detailed information for the SKILL record type
DISP S IEMP	displays detailed information for the set IEMP
DISP A	displays summary information for all areas
DISP I DEPT, LOCATION	displays detailed information for the LOCATION data item within the DEPT record type
DISP L	displays the language used as a basis for sorting

Sample session using DISPLAY with command arguments:

```
I: disp r
        record name is SYSTEM
                within (JOBS)
                (no title for this record)
                access is READ and WRITE
                participates in $SYSSET, IEMP, IDEP, IJOB
                (no items for this record)

        record name is DEPT
                within (JOBS)
                calc key is NAME
                duplicates are NOT ALLOWED
                title is "this is a title"
                access is READ and WRITE
                participates in $SYSSET, HAS, IDEP
                item name is DNUMBER     INT 1
                item name is NAME  CHAR 12
                item name is LOCATION     STR 35
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Display Command

```
record name is EMPLOYEE
        within (JOB1)
        calc key is ID
        duplicates are NOT ALLOWED
        (no title for this record)
        access is READ and WRITE
        participates in $SYSSET, IEMP, POSSESS, DETAILS, FILLEDBY, HAS
        item name is ID        CHAR 11
        item name is LASTNAME    CHAR 20
        item name is FNAME STR 12
        item name is PASTJOB       STR 25
        occurs 3 times
        item name is YTDEARN       IDEC 7, 2

record name is BIOGRAPH
        within (JOB2)
        (no title for this record)
        access is READ and WRITE
        participates in $SYSSET, DETAILS
        item name is LINE     STR 50
        occurs 5 times

record name is SKILL
        within (JOBS, JOB2)
        calc key is SKILCODE
        duplicates are NOT ALLOWED
        (no title for this record)
        access is READ and WRITE
        participates in $SYSSET, POSSESS, NEEDS
        item name is SKILCODE      UNSIGNED 2
        item name is DESCRIPT      STR 55
        item name is RATINGREAL 2

    record name is JOB
        within any area
        (no title for this record)
        access is READ and WRITE
        participates in $SYSSET, FILLEDBY, NEEDS, IJOB
        item name is JOBCODE       UNSIGNED 1
        item name is DESCRIPT      STR 50

I: disp r skill
    record name is SKILL
    within (JOBS, JOB2)
    calc key is SKILCODE
    duplicates are NOT ALLOWED
    (no title for this record)
    access is READ and WRITE
    participates in $SYSSET, POSSESS, NEEDS
    item name is SKILCODE      UNSIGNED 2
    item name is DESCRIPT      STR 55
    item name is RATINGREAL 2
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Display Command

```
I: disp s iemp
        set name is IEMP
                type is 1:N
                retention is FIXED
                (no title for this set)
                access is READ and WRITE
                owner is SYSTEM
                insertion is MANUAL
                order is IMMATERIAL
                member is EMPLOYEE
                insertion is AUTO
                order is SORTED by az LASTNAME , az FNAME
                duplicates are IMMATERIAL

I: disp a
        area name is JOBS
                file name is jobs.db
                size is 50 pages
                page size is 512
                pointers are ALLOWED
                (no title for this area)

        area name is JOB1
                file name is job1.db
                size is 20 pages
                page size is 512
                pointers are ALLOWED
                (no title for this area)

        area name is JOB2
                file name is job2.db
                size is 20 pages
                page size is 512
                pointers are ALLOWED
                (no title for this area)

I: disp i dept location
        item name is LOCATION      STR 35
                range is A to ZZZZZZ
                (no title for this item)
                access is READ and WRITE
```

© Copyright 1987 Micro Data Base Systems, Inc.

III. INVOKING DML COMMANDS INTERACTIVELY

SYNTAX

<dml command name><command string>

SYNOPSIS

A DML command can be invoked interactively by stating its name and its command string. Just as with MDBS DMS, a command string consists of from zero to three arguments indicating the data item, record type, set and/or area names that are pertinent to the DML command with which the command string is being used (see Appendix C). However, in IDML some DML utility (e.g., AUI) and assignment (e.g., SUM) commands also require numbers in their command strings. If a DML command needs input data to store into the data base, then the user is prompted for the needed data. If a DML command extracts data from the data base, that data is displayed at the console.

A ? facility is provided to obtain the command string form for any DML command; it also indicates whether data will be passed to or from the data base. If a command status other than 0 results from executing a DML command, then a command status message is displayed at the console.

DESCRIPTION

To execute a DML command interactively, simply enter its name and an appropriate command string. DML command names that can be used within IDML are presented in Appendices A and B. The valid command string entries for each command are shown in Appendix C. A blank space or comma can be used to separate a command name from its command string. Entries within a command string are separated by either blanks or commas. When accepting a DML command, IDML translates the command string to upper case. Thus upper and lower case can be intermixed when stating a DML command string.

Command strings can contain area names, set names, data item names and numbers (i.e., integers). In addition to consulting Appendix C, a user can also find out about the command string form for a given DML command by entering the DML command name followed by **?**. For instance, to find out the command string composition for SMM, enter:

smm?

IDML responds with the following message that indicates that two set names are needed in the SMM command string.

syntax of SMM is: SMM set set

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Invoking DML Commands

To see the command string for SMU, enter:

smu?

IDML responds with the following message that indicates the first entry in the command string is a set name and the second entry is a number (in this case the number of a user-defined currency indicator).

syntax of SMU is: SMU set number

If a command involves data input to or data output from the data base, then the ? facility will show that a data transfer is involved. For instance, ask for the syntax of the crs command:

crs?

The following reply shows the command string (a record name). The word **data** indicates that CRS transfers data between IDML and the data base control system. The brackets ([]) surrounding **data** indicate that the command line does not include data values.

syntax of CRS is: CRS record [data]

IDML prompts the user for the needed data values when it expects data input for a DML command. For example, to create an occurrence of the EMPLOYEE record type, the following interaction would be used (user responses are in italics).

```
I: crs,employee
        enter ID: 923487962
        enter LASTNAME: Smith
        enter FNAME: Bob
        enter PASTJOBS: Marketing
        enter PASTJOBS: Sales
        enter PASTJOBS:
        enter YTDEARN: 48000
```

The user terminates the entry of each data value by pressing the carriage return key. Notice that since PASTJOBS is a repeating data item (that occurs three times), the user is given three opportunities for specifying PASTJOBS values. In this example, the user specified only two of the three possible values.

The data modification commands prompt the user for each data item value, but additionally IDML provides the user with the current value surrounded by brackets, like this [current value]. To keep the current value of the data item, just press the carriage return.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Invoking DML Commands

When entering data for non-numeric data items, IDML reads in characters up to the maximum size of the data item with blank fill on the right. If something other than a blank is desired for character data items, the FILLCHR command can be used to designate a non-blank symbol as the fill character. When FILLCHR is invoked it should be followed by either the desired symbol enclosed in a pair of quotes or an integer whose ASCII character equivalent will serve as the fill character.

The following example causes subsequent values entered for character data items to be filled with the null character.

```
FILLCHR 0
```

To later revert back to blank fill, use the following command:

```
FILLCHR " "
```

When entering data for a numeric data item, IDML issues the DMS command status of 33 if the value is out of the maximum possible range allowed for that data item's size. If any value being entered is outside of the feasible range of values for the data item (if a feasibility range is specified in the DDL), then the DMS command status of 30 is returned. When prompted for a data value, pressing the carriage return key will be interpreted as a 0 for numeric data items and a blank for other data items.

If a non-zero command status results from interactively executing a DML command, then that command status is displayed at the console and the I: prompt is issued.

When data is retrieved by a command, that data is displayed at the console. IDML prefaces each data value displayed with its corresponding data item name. For instance, to see the data that was just loaded by the above CRS command, use the GETC command:

```
I: getc
        ID:  923487962
        LASTNAME:  Smith
        FNAME:  Bob
        PASTJOBS:  Marketing
        PASTJOBS:  Sales
        PASTJOBS:
        YTDEARN:  48000
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Invoking DML Commands

Several utility commands (aui,ccu,dbenv), as well as assignment commands involving user-defined currency indicators (scu, smu, sou, suc, sum, sun, suo, suu), must have numbers specified in their command strings. With MDBS DMS these numbers were not part of command strings, but were passed with data blocks or program record types. The use of a number within a command string is illustrated by:

```
I: sum?
        syntax for SUM is:  SUM number set
I: sum 7 set3
```

The result is that user indicator 7 is assigned to be the same as the current member of SET3.

If a set is omitted from a command string, it defaults to the last set name specified in a command string. The same is true for areas, data items, record types, and numerical values. As an example, suppose that we want to scroll through all department names:

```
I: ffm?
        syntax for FFM is:  FFM set
I: gfm?
        syntax for GFM is:  GFM item set [data]
I: ffm idep
I: gfm name
        NAME:  cust support
I: fnm
I: gfm
        NAME:  production
I: fnm
I: gfm
        NAME:  R & D
        .
        .
        .
I: fnm
        DMS command status 255
```

Notice that in the first usage of GFM the set name was not specified, so it defaulted to the last set name that had previously been specified (i.e., idep). The same is true for the usage of FNM. In the second (and subsequent) usage of GFM the data item name was not specified, so it defaulted to the last item name that had previously been specified (i.e., name). In commands where two sets are required, the last set is the one saved.

For example, the following command sets the default set type (remembered by IDML) to idep. The same convention holds if two numbers are required for a command string:

```
som has idep
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Invoking DML Commands

EXAMPLES (user responses are in italics)

```
I:  crs?
        syntax for CRS is:  CRS record [data]
I:  crs dept
        enter DNUMBER:  14
        enter NAME:  personnel
        enter LOCATION:  Martinsville
I:  ims?
        syntax for IMS is:  IMS set
I:  crs employee
        enter ID:  123456777
        enter LASTNAME:  Moore
        enter FNAME:  Steve
        enter PASTJOBS:  clerk
        enter PASTJOBS:
        enter PASTJOBS:
        enter YTDEARN:  58392
I:  ims has
I:  crs employee
        enter ID:  923487962
        enter LASTNAME:  Smith
        enter FNAME:  Bob
        enter PASTJOBS:  Marketing
        enter PASTJOBS:  Sales
        enter PASTJOBS:
        enter PASTJOBS:
        enter YTDEARN:  48000
I:  ims has
I:  crs biograph
        enter LINE:  Birthdate 3/21/28.  Early years in Chicago area.
        enter LINE:  Graduated from Purdue University 1949.
        enter LINE:
        enter LINE:
        enter LINE:
        .
        .
        .
I:  fmsk?
        syntax for FMSK is:  FMSK set [data]
I:  fmsk iemp
        enter LASTNAME:  Moore
        enter FNAME:  Bob

        DMS command status 255
I:  fmsk
        enter LASTNAME:  Moore
        enter FNAME:  Steve
I:  gfc ytdearn
        YTDEARN:  58392
I:  smm has iemp
I:  geto has
        DNUMBER:  14
        NAME:  personnel
        LOCATION:  Martinsville
I:  quit
```

© Copyright 1987 Micro Data Base Systems, Inc.

IV. DEFINE COMMAND

SYNTAX

<define>

SYNOPSIS

The DEFINE command can be used to define macros (i.e., new, user- defined commands or parts of commands). These definitions are held in the data base and therefore exist across QRS and IDML sessions. The definition process itself is interactive and menu-driven.

A macro is a word that can be equated to any section of an IDML command and can be freely substituted for that portion of the command. It can also be equated to an entire IDML command or to a sequence of IDML commands (involving many lines of input). Thus macros provide a convenient means for storing and later using often-used IDML commands or sections of a command. A macro for a series of IDML commands is especially useful for iteratively executing the same DML logic. The user is prompted for any input needed by an IDML command within the macro. If a command within a macro returns command status of 255, the status is displayed and macro execution continues. Any other non-zero command status halts the macro.

A macro defined with IDML can later be used within QRS. A macro defined with QRS can later be used within IDML, provided its text consists of a valid command expression within IDML. Parameterized macros are supported in IDML. When a macro is to be used for any part of a DML command string, that macro's name must be in upper case.

Macros may not be nested within other macros. Macros may, however, be included within a command file executed via the READ command (see Chapter V). Because synonyms are treated very similarly to macros, a synonym will not be recognized within macro text.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Define Command

DESCRIPTION

When the DEFINE command is given, the following menu is displayed:

Macro Definition Utility

Macro/Synonym Functions:
(A) Add a Macro
(C) Change a Macro
(D) Delete a Macro
(I) Index of Macro/Synonym Names
(L) List Macros and Synonyms
(?) Print This Command List
(Q) Quit Define Mode

Macro/Synonym Function?

The effects of these seven macro definition functions are described below.

A: The user is prompted for the name of the new macro and for the IDML text that it will represent. A macro name must be alphanumeric, and cannot begin with a digit. Use of this macro name within an IDML command will cause the macro's text to be substituted for the macro name. The text for a macro plus the length of the macro name cannot exceed 255 characters. Macros occurring within the text of another macro are not recognized. Synonyms occurring within the text of a macro are not recognized. The macro is not translated to upper case; it must be referenced in the same case with which it is defined in this function.

C: The user is prompted for the name of an existing macro and given the choice of changing the macro name or the macro text. DEFINE prompts the user for the new name or new macro text, depending upon the response of the user.

D: The user is prompted for the name of the macro to be deleted.

I: An index of all macros and synonyms is displayed. Synonyms are indented, macros are not.

L: The user is prompted for a macro or synonym name. Pressing the return key will result in a listing of all macros (and their text) and all synonyms. If the user enters the name of a particular macro, its text is displayed. If the user enters the name of a particular synonym, then the name of the area, set, record type or data item for which it is a synonym is displayed. A user may not define a macro that has the same name as a synonym.

?: The macro definition functions are displayed.

Q: The user leaves the macro definition mode. IDML gives the I: prompt.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Define Command

EXAMPLES (user responses are in italics)

I: ***define***

Macro Definition Utility

Macro/Synonym Functions:
(A) Add a Macro
(C) Change a Macro
(D) Delete a Macro
(I) Index of Macro/Synonym Names
(L) List Macros and Synonyms
(?) Print This Command List
(Q) Quit Define Mode

Function?*A*
Name?*CEMP*

Enter text below (terminate with an empty line)
******fmi jobcode ijob*
******som filledby ijob*
******crs emp*
******crs biograph*
******fmi dnumber idep*
******som has idep*
******ims has*
*****

Function?*A*
Name?*N*

Enter text below (terminate with an empty line)
******fnm*
******getc*
*****

Function?*I*
DNO (This is a synonym; it is indented)
SC
CEMP (This is a macro; it is not indented)
N

Function?*C*
Name?*N*

Enter text below (terminate with empty line)
******getc*
******fnm*
*****

Function?*Q* (leave DEFINE)

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Define Command

```
I:  CEMP                                        (return to IDML from DEFINE)
        enter JOBCODE:  5
        enter ID:  123443210
        enter LASTNAME:  Ferguson
        enter FNAME:  Kristina
        enter PASTJOBS:  sales rep
        enter PASTJOBS:
        enter PASTJOBS:
        enter YTDEARN:  29840
        enter LINE:  Date of birth 12/13/51.
        enter LINE:  Background in medicine and marketing.
        enter LINE:  Salesperson of the year in 1979.
        enter LINE:
        enter LINE:
        enter DNUMBER:  7
I:  CEMP
        .
        .
        .
I:  ffm idep

I:  N
            DNUMBER:  3
            NAME:  cust support
            LOCATION:  Indianapolis
I:  N
            DNUMBER:  5
            NAME:  production
            LOCATION:  Lafayette
I:  N
            .
            .
            .
I:  N
            DNUMBER:  13
            NAME:  sales
            LOCATION:  Lafayette
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Define Command

ADVANCED USAGE OF MACROS WITH PARAMETERS

When specifying the text of a macro with the DEFINE command, up to 9 parameters can be embedded within the macro text. Each macro parameter functions as a variable whose value can be stated by a user whenever the macro is executed. For example, the text of a macro called SQ might be defined as:

```
SOM &1,&2
```

The macro parameters are &1 and &2. If a user is interested in setting the current owner of POSSESS to be the same as the current member of HAS, then the macro would be executed as:

```
SQ(POSSESS,HAS)
```

If the user is later interested in setting the current owner of NEEDS to be the same as the current member of IJOB, then the same macro is executed, but with different parameter values:

```
SQ(NEEDS,IJOB)
```

As the above examples show, IDML substitutes the values specified by a user for the parameters declared in the original macro text. The first value appearing in the parentheses following the macro name is substituted for &1, the second value is substituted for &2, and so forth.

As a parameter is specified in macro text, it can be defined to be an optional parameter. When a user executes a macro, no value needs to be indicated for an optional parameter. A parameter is defined to be optional by appending the parameter with the = symbol followed by a space. For example, if the macro text for GP is:

```
GETM &1, &2=
```

then the second parameter is optional. When executing the macro, a value for the second parameter may be omitted:

```
GP(IJOB,)
```

or it may be included:

```
GP(IJOB,PUTC)
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Define Command

Default values for parameters can also be specified in the macro text. For instance, if the macro text for SQ is:

```
SOM &1, &2=HAS
```

then HAS is the default value for this macro's second parameter. Whenever a user executes SQ without specifying a value for the second parameter, IDML uses HAS as the value. For instance:

```
SQ(DETAILS,)
```

sets the current owner of DETAILS to be the same as the current member of HAS. If a user explicitly states a value for the second parameter, then IDML ignores the default value.

A parameter can be used more than once within a macro's text. As an example, the macro text:

```
FMSK &1
PFM &2,&1
```

finds a record and then gives one of its fields a new value. If the name of this macro is PMSK, then:

```
PMSK (IEMP,LASTNAME)
```

finds a member of IEMP based on a sort key value and changes that member's last name.

Rules for Specifying Parameters in Macro Text

- &1, &2, ..., &9 are valid macro parameters. If n parameters are specified, they should be &1, .. &n (where $1 \leq n \leq 9$); &1 must be the first parameter used in the macro text; the remaining parameters can be specified in any order within the macro text.
- Any parameter can appear more than once in a macro's text.
- To signify that a parameter is optional, an equal sign and space must immediately follow the parameter number (e.g., &5=) whenever it appears in the macro text.
- To specify a default value for a parameter, the parameter number is immediately followed by an equal sign and the desired default value. Any letter or digit can appear in a default value. Any of the following characters is also allowed:

  " [ ] . / * + - < > = # $

  No spaces, commas, or parentheses are allowed in the default value unless they are within a matching pair of double quotes. A parameter or macro cannot appear as part of a default value. If a parameter appears multiple times in a macro's text, it can have a different default value at each place it appears.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Define Command

Rules for Running a Parameterized Macro

- The macro name is followed immediately (no space) by a matching set of parentheses containing the desired values. The values are separated by commas.
- Macro evaluation is based on positional substitutions. IDML substitutes the first value in the parentheses for &1 (the leftmost parameter), the second value is substituted for &2 (the next parameter), and so forth. If no value is given for a parameter (i.e., there are two consecutive commas), then the parameter's default value is used. If it does not have a default value, but is optional, then the parameter is ignored. Omitting a value for a required parameter results in an error message.
- The number of commas in a macro's list of values must always be one less than the number of parameters. If too many or too few values are specified, an error message results.
- A parameter value cannot contain spaces, commas, or parentheses unless they are enclosed in a matching pair of double quotes. A macro cannot be used as a parameter value. In addition to digits and numbers, a value can contain any of the following characters:

" [ ] . / * + - < > = # $

© Copyright 1987 Micro Data Base Systems, Inc.

V. READ COMMAND

SYNTAX

<read><*file–name*><batch>

SYNOPSIS

The READ command causes IDML to perform a series of IDML commands from a specified disk file. The file-name clause indicates the file containing the IDML commands. DML commands that require data prompt the user for the information.

The **batch** option instructs IDML to process the input file in batch mode; IDML assumes the input file contains DML commands and the corresponding data values for each DML command. While processing the input file, IDML expects no user input.

DESCRIPTION

In using the READ command, the indicated file name must be quoted and must be fully qualified within the host operating system. The file format is ASCII text, created with any standard text editor. Each line contains a valid IDML command (except READ) or a data value.

For example, the READ command can perform a series of macros. If an error is encountered, the READ command terminates; IDML does not process any remaining lines from the input file.

If **batch** processing is requested, each required input for a DML command will be read from succeeding lines in the file. Thus for batch processing, the file used by READ should contain data as well as DML commands.

For example, assume the input file for a read command includes the following three lines:

```
CRS JOB
125
President
```

IDML must process this file with the batch option and will not prompt the user for data values. This example results in a new JOB record occurrence whose code is 125 and whose description is President.

EXAMPLES

```
READ "QFILE"
READ "B:BATCH.TXT" BATCH
```

© Copyright 1987 Micro Data Base Systems, Inc.

VI. ECHO COMMAND

SYNTAX

<echo><*message*>

SYNOPSIS

An ECHO command displays a specified message on the console. By embedding ECHO commands within the input file for a READ command, an ongoing progress report of the READ command processing can be provided to the user.

DESCRIPTION

ECHO commands can be used in a READ command's file, in a startup file (Chapter X) or within a macro (Chapter IV) to cause messages to appear on the console screen. The message following an **echo** keyword will be output to the next line on the screen. A message can be either a number or a string of characters enclosed in a matching pair of quotes. If there is no message, a blank line is output.

EXAMPLES

```
ECHO 29
ECHO "The instructions are as follows:"
ECHO
ECHO 'Processing is complete'
```

© Copyright 1987 Micro Data Base Systems, Inc.

VII. COMMENT COMMAND

SYNTAX

<comment><*symbol*>

SYNOPSIS

The COMMENT command designates the character that IDML interprets as the beginning of a comment. Comments can be included for documentation purposes in a file processed by READ. The default comment character is the semicolon (;).

DESCRIPTION

Whenever the comment character appears in a command, the remainder of the line is considered to be a comment and therefore is ignored by the IDML processor. The COMMENT command can change the comment character to another character by including the new symbol with the command. The symbol should be enclosed in quotes following the **comment** keyword.

EXAMPLE

```
COMMENT " : "
```

© Copyright 1987 Micro Data Base Systems, Inc.

VIII. SET COMMAND

SYNTAX

<set opt><number><status>

or

<set opt><number>

SYNOPSIS

The SET command controls the environment settings available within IDML. The environment settings are options that have two statuses, **on** or **off**. The SET command changes the status of an environment option.

DESCRIPTION

The set command functions consistently with IDML and QRS. When setting an option, specify the option number and the desired status. Also, you can toggle the current setting for an environment option by omitting the status from the command; that is, if the option is on, IDML turns it off and vice-a-versa.

The following summary lists the available environment options and the default status (on or off) of each. A full description of each is provided after this summary.

Option number	Default status	Meaning if **on**
5	**OFF**	echo alternate input
11	**OFF**	suppress macro recognition
12	**OFF**	echo macro expansion output
15	**OFF**	pause after each report page

Environment Option Details

OPT 5 Echo Alternate Input **OPT 5**

A sequence of IDML commands can be batched within a text file and then executed non-interactively with the READ command (Chapter VIII). To have the text file queries echoed to the console, this option should be on. The default status is off.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Set Command

OPT 11 Suppress Macro Recognition **OPT 11**

While this option is on, all macros that have been defined with the DEFINE command (see Chapter VII) are ignored by IDML. If this option is on when DEFINE is used to define a macro, keywords can be defined as macros. Those macros will later be recognized if this option is turned off. The default status is off.

OPT 12 Echo Macro Expansion Output **OPT 12**

When this option is on, every command is echoed to the console. If a command contains macros, the full expansions of those macros are inserted into the command echo. The default status is off. This command is particularly helpful when first becoming acquainted with macros.

OPT 15 Pause After Each Report Page **OPT 15**

When this option is on, IDML will pause at the end of each report page. Pressing any key will cause the next report page to be displayed. The default status is off.

EXAMPLES

```
SET OPT 5 ON
SET OPT 11 OFF
set opt 12
```


© Copyright 1987 Micro Data Base Systems, Inc.

IX. QUIT COMMAND

SYNTAX

<quit> or **<bye>**

SYNOPSIS

The QUIT command terminates an IDML session and automatically closes the data base.

DESCRIPTION

The quit command terminates IDML and returns the user to the operating system. QUIT closes the data base (saving all macro definitions in the main data base area). If a data base is not closed, it cannot be used in the future.

In addition to **quit, bye** may also be used to close the data base and return the user to the operating system.

Running the Quit successfully is indicated by the "data base closed" message.

© Copyright 1987 Micro Data Base Systems, Inc.

X. START-UP FILE

OVERVIEW

An IDML user can begin an IDML command session as described in Chapter I, Section B. Alternatively, a start-up file can be created which specifies the file name of the main data base area, user name, user password, and/or a series of IDML commands. When IDML is executed, its first activity is to read a file named **STARTUP.TXT** if one exists. If there is none, the IDML processor looks for a file named **STARTI** and treats it as the start-up file if it exists. If the main area file name, user name, or user password is absent from the start-up file, the IDML user is prompted for the missing value(s). Then, all IDML commands on the start-up file are executed. When these commands have been executed, the IDML I: prompt is displayed.

CREATING THE START-UP FILE

The start-up file can be created, edited, and saved by using the MDBS.DDL software or any other standard text editing facilities. The approach used is similar to that used for creating an input file of IDML commands for the READ command (Chapter VI).

The first line of the start-up file must begin with the word **START**. A file name (for a data base's main area), user name and/or user password can optionally be specified on this line. If present, they must appear in this order. For example:

```
START "JOBS.DB" "D LEHR" "smiles"
```

Here JOBS.DB is the fully qualified name of the file containing the main data base area, D LEHR is a user name, and "smiles" is the password. If any of these arguments is omitted from the first line of start-up, the user is prompted for the missing value(s). If the user fails to provide appropriate values, IDML returns control to the operating system. Each subsequent line of the start-up file can consist of any valid IDML command.

The start-up file must reside on the currently logged-on drive/directory and its name must be consistent with the operating system under which the IDML user is working.

In many operating systems, start-up information can be created on files other than the start-up file. In this case, the start-up information on such a file can be used in place of the start-up file by entering the file name on the operating system command line that is used to begin the execution of the IDML processor. See the pertinent system specific manual for details.

© Copyright 1987 Micro Data Base Systems, Inc.

XI. MESSAGES

These are messages that can occur in the interactive use of IDML. If a DMS command is being executed with IDML, then various DMS command status messages can result. These are described in the MDBS DMS Manual, and will always be prefaced with the phrase "DMS command status". The messages encountered within the IDML logic are described below.

*****Alternate input files may not be nested.**

Explanation:
A file cannot be read from within another file which is being read.
Possible Causes and/or Solutions:
1. A 'READ' command has been used on a file which itself contains a 'READ' command.

*****Data dictionary not accessible.**

Explanation:
The data dictionary cannot be accessed with the DISPLAY command.
Possible Causes and/or Solutions:
1. The data base was initialized with the **o** option.

*****Enclose file name in quotes.**

Explanation:
Because of non-alphanumeric characters which are allowed for file names, for many operating systems, IDML requires that file names be in quotes.
Possible Causes and/or Solutions:
1. The file name specified must be enclosed in quotes.

*****File not present.**

Explanation:
The file specified is not present on the disk.
Possible Causes and/or Solutions:
1. The system could not find the file indicated in the READ command.

*****Insufficient room in memory.**

Explanation:
There is not enough room in memory to carry out any further processing by IDML.
Possible Causes and/or Solutions:
1. There is not enough room in memory to open the data base.
2. If the data base has already been opened, there is insufficient memory to process the given command.
3. See the appropriate system specific manual.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Messages

*****Invalid area.**

Explanation:
Self-explanatory.
Possible Causes and/or Solutions:
1. The area name specified in an IDML command is not a valid area, as defined in the DDL specification.
2. Check list of valid areas using the DISP A command.

*****Invalid DISPLAY parameter.**

Explanation:
The parameter specified is not a valid DISPLAY parameter.
Possible Causes and/or Solutions:
The parameter specified is not one of a, i, l, r, s.

*****Invalid item name.**

Explanation:
The specified data item name does not exist for the record that is to be processed.
Possible Causes and/or Solutions:
1. The item name is misspelled or typed as record.item (which is invalid in IDML).
2. Check the command syntax for the proper sequence of arguments.

*****Invalid number.**

Explanation:
The indicated field contains an invalid number.
Possible Causes and/or Solutions:
1. An invalid character has been specified within a numeric field.
2. Check numeric specifications in query command.

*****Missing area.**

Explanation:
An expected area name is missing.
Possible Causes and/or Solutions:
1. An area was expected in the command string. It was missing and no area had been specified in any earlier IDML command.

*****Missing item.**

Explanation:
An expected data item name is missing.
Possible Causes and/or Solutions:
1. A data item was expected in the command string. It was missing and no item name had been specified in any earlier IDML command.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
Messages

*****Missing number.**

Explanation:
Self-explanatory.
Possible Causes and/or Solutions:
1. A number was expected in the command string. It was missing and no number had been specified in any earlier IDML command.

*****Missing record.**

Explanation:
An expected record type name is missing.
Possible Causes and/or Solutions:
1. A record type was expected in the command string. It was missing and no record type name had been specified in any earlier IDML command.

*****Missing set.**

Explanation:
An expected set name is missing.
Possible Causes and/or Solutions:
1. A set was expected in the command string. It was missing and no set name had been specified in any earlier IDML command.

*****Syntax error-command expected.**

Explanation:
The system was expecting a command name and encountered a non- alphanumeric character within the command name.
Possible Causes and/or Solutions:
1. A character within the command name was not alphanumeric.

*****Unmatched quotes.**

Explanation:
Self-explanatory.
Possible Causes and/or Solutions:
1. Quotes have been improperly specified or improperly matched.

© Copyright 1987 Micro Data Base Systems, Inc.

APPENDIX A

DML COMMAND GROUPS

MDBS IDML MANUAL
DML Command Groups

APPENDIX A

ASSIGNMENT COMMANDS

DMS Reference Manual
Chapter VI

SCD	Set Current of run unit to Data base key
SCM	Set Current of run unit based on Member
SCN	Set Current of run unit to Null
SCO	Set Current of run unit based on Owner
SCU	Set Current of run unit based on User indicator
SDC	Save Data base key for Current of run unit
SMC	Set Member based on Current of run unit
SME	Set Member to current of run unit Exception
SMM	Set Member based on Member
SMN	Set Member to Null
SMO	Set Member based on Owner
SMU	Set Member based on User indicator
SOC	Set Owner based on Current of run unit
SOE	Set Owner to current of run unit Exception
SOM	Set Owner based on Member
SON	Set Owner to Null
SOO	Set Owner based on Owner
SOU	Set Owner based on User indicator
SUC	Set User indicator to Current of run unit
SUM	Set User indicator to Member
SUN	Set User indicator to Null
SUO	Set User indicator to Owner
SUU	Set User indicator to User indicator

BOOLEAN COMMANDS

DMS Reference Manual
Chapter XIII

AMM	And of Members with Members
AMO	And of Members with Owners
AOM	And of Owners with Members
AOO	And of Owners with Owners
XMM	eXclude Members from Members
XMO	eXclude Members from Owners
XOM	eXclude Owners from Members
XOO	eXclude Owners from Owners

CONNECT COMMANDS

DMS Reference Manual
Chapter VIII

IMS	Insert Member into Set
IOS	Insert Owner into Set

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
DML Command Groups

CREATION COMMANDS

DMS Reference Manual
Chapter VII

CRA	Create Record in Area
CRS	Create Record and Store

DELETION COMMANDS

DMS Reference Manual
Chapter X

DRC	Delete Record that is Current
DRM	Delete Record that is Member
DRO	Delete Record that is Owner

DISCONNECT COMMANDS

DMS Reference Manual
Chapter IX

RMS	Remove Member from Set
ROS	Remove Owner from Set
RSM	Remove all Set Members
RSO	Remove all Set Owners

FIND COMMANDS

DMS Reference Manual
Chapter III

FDRK	Find Duplicate Record based on calc Key
FFM	Find First Member
FFO	Find First Owner
FFS	Find First Sequential record
FLM	Find Last Member
FLO	Find Last Owner
FMI	Find Member based on data Item
FMSK	Find Member based on Sort Key
FNM	Find Next Member
FNMI	Find Next Member based on data Item
FNMSK	Find Next Member based on Sort Key
FNO	Find Next Owner
FNOI	Find Next Owner based on data Item
FNOSK	Find Next Owner based on Sort Key
FNS	Find Next Sequential record
FOI	Find Owner based on data Item
FOSK	Find Owner based on Sort Key-
FPM	Find Prior Member
FPMI	Find Prior Member based on data Item
FPMSK	Find Prior Member based on Sort Key
FPO	Find Prior Owner
FPOI	Find Prior Owner based on data Item
FPOSK	Find Prior Owner based on Sort Key
FRK	Find Record based on calc Key

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
DML Command Groups

MODIFY COMMANDS

DMS Reference Manual
Chapter V

PFC	Put data into Field of Current of run unit
PFM	Put data into Field of Member
PFO	Put data into Field of Owner
PUTC	PUT data into Current of run unit
PUTM	PUT data into Member
PUTO	PUT data into Owner

MULTIUSER COMMANDS

DMS Reference Manual
Chapter XV

MAU	Multiuser Active User indicators
MCC	Multiuser Contention Count
MCF	Multiuser Current of run unit Free
MCP	Multiuser Current of run unit Protect
MRID	Multiuser Run unit IDentifiaction
MRTF	Multiuser Record Type Free
MRTP	Multiuser Record Type Protect
MSF	Multiuser Set Free
MSP	Multiuser Set Protect

RETRIEVAL COMMANDS

DMS Reference Manual
Chapter IV

GETC	GET data from Current of run unit
GETM	GET data from Member
GETO	GET data from Owner
GFC	Get Field from Current of run unit
GFM	Get Field from Member
GFO	Get Field from Owner
ODRK	Obtain Duplicate Record based on calc Key
OFM	Obtain First Member
OFO	Obtain First Owner
OLM	Obtain Last Member
OLO	Obtain Last Owner
OMI	Obtain Member based on data Item
OMSK	Obtain Member based on Sort Key
ONM	Obtain Next Member
ONMI	Obtain Next Member based on data Item
ONMSK	Obtain Next Member based on Sort Key
ONO	Obtain Next Owner
ONOI	Obtain Next Owner based on data Item
ONOSK	Obtain Next Owner based on Sort Key
OOI	Obtain Owner based on data Item
OOSK	Obtain Owner based on Sort Key

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS IDML MANUAL
DML Command Groups

RETRIEVAL COMMANDS (CONT.)

DMS ReferenceManual
Chapter IV

OPM	Obtain Prior Member
OPMI	Obtain Prior Member based on data Item
OPMSK	Obtain Prior Member based on Sort Key
OPO	Obtain Prior Owner
OPOI	Obtain Prior Owner based on data Item
OPOSK	Obtain Prior Owner based on Sort Key
ORK	Obtain Record based on calc Key

UTILITY COMMANDS

DMS Reference Manual
Chapter XII

AUI	Allocate User Indicators
CCU	Check Current of run unit against User indicator
DBCLS	Data Base CLoSe
DBCLSA	Data Base CLoSe for Area
DBCNV	Data Base format CoNVersion
DBENV	Data Base ENVironment
DBOPNA	Data Base OPeN Area
DBSAVE	Data Base SAVE
DBSTAT	Data Base STATistics
GMC	Get Member Count
GOC	Get Owner Count
GTC	Get Type of Current of run unit
GTM	Get Type of Member
GTO	Get Type of Owner
NCI	Null all Currency Indicators
TCN	Test Current of run unit for Null
TCT	Test Current of run unit Type
TMT	Test Member Type
TOT	Test Owner Type

© Copyright 1987 Micro Data Base Systems, Inc.

APPENDIX B
DML COMMAND STRING FORMS

MDBS IDML MANUAL
DML Command Forms

APPENDIX B

amm, set, set, set
amo, set, set, set
aom, set, set, set
aoo, set, set, set
aui, number
ccu, number
cra, record, area, data
crs, record, data
dbclsa, area
dbcnv, data
dbenv, data
dbopna, area, data
dbsave
dbstat, data
drc
drm, set
dro, set
fdrk, record, data
ffm, set
ffo, set
ffs, area
flm, set
flo, set
fmi, item, set, data
fmsk, set, data
fnm, set
fnmi, item, set, data
fnmsk, set, data
fno, set
fnoi, item, set, data
fnosk, set, data
fns, area
foi, item, set, data
fosk, set, data
fpm, set
fpmi, item, set, data
fpmsk, set, data
fpo, set
fpoi, item, set, data
fposk, set, data
frk, record, data
getc, data
getm, set, data
geto, set, data
gfc, item, data
gfm, item, set, data
gfo, item, set, data
gmc, set, data
goc, set, data
gtc, data
gtm, set, data
gto, set, data
ims, set
ios, set
lgflsh
lgmsg, data
mau, data
mcc, data
mcf
mcp
mrid, data
mrtf, record
mrtp, record
msf, set
msp, set
nci
odrk, record, data
ofm, set, data
ofo, set, data
olm, set, data
olo, set, data
omi, item, set, data
omsk, set, data
onm, set, data
onmi, item, set, data
onmsk, set, data
ono, set, data
onoi, item, set, data
onosk, set, data
ooi, item, set, data
oosk, set, data
opm, set, data
opmi, item, set, data
opmsk, set, data
opo, set, data
opoi, item, set, data
oposk, set, data
ork, record, data
pfc, item, data
pfm, item, set, data
pfo, item, set, data
putc, data
putm, set, data
puto, set, data
rms, set
ros, set
rsm, set
rso, set
scd, data
scm, set
scn, set
sco, set
scu, number
sdc, data
smc, set
sme, set
smm, set, set
smn, set
smo, set, set
smu, set, number
soc, set
soe, set
som, set, set
son, set
soo, set, set
sou, set, number
suc, number
sum, set, number
sun, number
suo, set, number
suu, number, number
tcn
tct, record
tmn, set
tmt, record, set
ton, set
tot, record, set
tun, number
xmm, set, set, set
xmo, set, set, set
xom, set, set, set
xoo, set, set, set

© Copyright 1987 Micro Data Base Systems, Inc.

DOCUMENTATION COMMENT FORM

MDBS Document Title: _____ Version: _____

We welcome and appreciate all comments and suggestions that can help us to improve our manuals and products. Use this form to express your views concerning this manual.

Please do not use this form to report system problems or to request materials, etc. System problems should be reported to MDBS by phone or telex, or in a separate letter addressed to the attention of the technical support division. Requests for published materials should be addressed to the attention of the marketing division.

Sender:

_____ (name)

_____ (company)

_____ (position)

_____ (address)

_____ (telephone)

_____ (address)

_____ (city, state, zip)

COMMENTS:

Areas of comment are general presentation, format, organization, completeness, clarity, accuracy, etc. If a comment applies to one or more specific pages, please cite the page numbers.

_____

_____

_____

_____

_____

_____

_____

Continue on additional pages, as needed. Thank you for your response.

© COPYRIGHT 1987 Micro Data Base Systems, Inc.